Evolving Circuits in Seconds: Experiments with a Stand-Alone Board-Level Evolvable System

Adrian Stoica Ricardo S. Zebulum M. I. Ferguson

Didier Keymeulen Vu Duong

Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA 91109
adrian.stoica@jpl.nasa.gov

Xin Guo
Chromatech, Alameda CA 94501

Abstract

The purpose of this paper is twofold: first, to illustrate a stand-alone board-level evolvable system (SABLES) and its performance, and second to illustrate some problems that occur during evolution with real hardware in the loop, or when the intention of the user is not completely reflected in the fitness function. SABLES is part of an effort to achieve integrated evolvable systems. SABLES provides autonomous, fast (tens to hundreds of seconds), on-chip evolution involving about 100,000 circuit evaluations. Its main components are a JPL Field Programmable Transistor Array (FPTA) chip used as transistor-level reconfigurable hardware, and a TI DSP that implements the evolutionary algorithm controlling the FPTA reconfiguration. The paper details an example of evolution on SABLES and points out to certain transient and memory effects that affect the stability of solutions obtained reusing the same piece of hardware for rapid testing of individuals during evolution. It also illustrates how specifications not completely reflected in the fitness function, such as the time scales of response for logical circuits, may lead to overall unsatisfactory solutions. Both such situations can be handled with appropriate modification of fitness function and additional testing.

1. Introduction

An evolvable hardware system is constituted of two main components: the reconfigurable hardware (RH) and the reconfiguration mechanism (RM). Figure 1 illustrates several ways of implementing the two components. In previously reported research the evolutionary processor (EP) that acts as a RM was implemented on a variety of platforms including supercomputer [1] [2], single PC (most of researchers, see e.g. [3]), DSP [4] FPGA [5] and ASIC [6]. The RH was approached as simulated model of unconstrained topology [7,8,9], real FPGA [10,11], FPAA model [12] or actual chips[13], FPTA model [14] or actual chip[15].

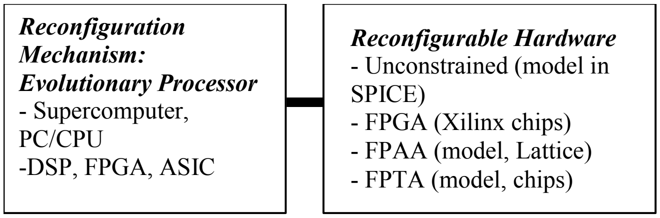
Reconfiguration Mechanism: Evolutionary Processor
- Supercomputer, PC/CPU
-DSP, FPGA, ASIC
Reconfigurable Hardware
- Unconstrained (model in SPICE)
- FPGA (Xilinx chips)
- FPAA (model, Lattice)
- FPTA (model, chips)

Figure 1. A block diagram of an EHW system with several means of implementing the reconfiguration mechanism and the reconfigurable hardware.

Real-world applications will require compact, low-power, autonomous evolvable hardware. A discussion on the evolution of the integration in Evolvable Systems was presented in [16]. An effort in transitioning from PC -simulated or PC-controlled evolutions to embedded and ultimately to integrated system-on-a-chip evolvable systems is needed. Pioneering efforts in this direction include the work of Higuchi who implemented a RM (Genetic Algorithm) chip connected to prosthetic hand

[17], and Shakelford who has implemented a digital system [5].

This paper describes the results of such an integrated effort. The SABLES solution provides autonomous, fast (about 1,000 circuit evaluations per second), on-chip circuit reconfiguration. Its main components are a JPL Field Programmable Transistor Array (FPTA) chip as transistor-level reconfigurable hardware, and a TI DSP implementing the evolutionary algorithm as the controller for reconfiguration. SABLES achieves approximately 1-2 orders of magnitude reduction in memory and about 4 orders of magnitude improvement in speed compared to systems evolving in simulations, and about 1 order of magnitude reduction in volume and 1 order of magnitude improvement in speed (through improved communication) compared to a PC controlled system using the same FPTA chips.

The paper has two parts. Section 2 overviews the components of SABLES, including the FPTA2 chip and the DSP system. The evolution of a half-wave rectifier circuit is presented to illustrate how the system functions. Section 3 concentrates on some stability and reproducibility aspects of solutions evolved by rapid testing of candidate solutions on the same piece of hardware. It also emphasizes the importance of having implicit assumptions explicitly reflected/incorporated in the fitness function. For example, it can be enforced that a logic gate has the same response at different time scales by applying a method derived as a form of mixtrinsic evolution [18] (using a mixed population of individuals tested at two different time scales).

2. A stand-alone board-level evolvable system

2.1 SABLES components

SABLES integrates an FPTA and a DSP implementing the Evolutionary Platform (EP) as shown in Figure 2. The system is stand-alone and is connected to the PC only for the purpose of receiving specifications and communicating back the results of evolution for analysis.

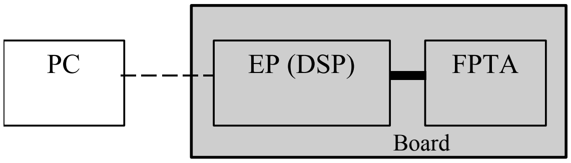

Figure 2 Block diagram of a simple stand-alone evolvable system.

The FPTA is an implementation of an evolution-oriented reconfigurable architecture (EORA) [15]. The lack of evolution-oriented devices, in particular for analog, has been an important stumbling block for researchers attempting evolution in intrinsic mode (with evaluation directly in hardware). Extrinsic evolution (using simulated models) is slow and scales badly when performed accurately e.g. in SPICE, and less accurate models may lead to solutions that behave differently in hardware than in software simulations. The FPTA has transistor level reconfigurability, supports any arrangement of programming bits without danger of damage to the chip (as is the case with some commercial devices). Three generations of FPTA chips have been built and used in evolutionary experiments. The latest chip, FPTA-2, consists of an 8x8 array of reconfigurable cells. Each cell has a transistor array as well as a set of other programmable resources, including programmable resistors and static capacitors. Figure 3 provides a broad view of the chip architecture together with a detailed view of the reconfigurable transistor array cell. The reconfigurable circuitry consists of 14 transistors connected through 44 switches and is able to implement different building blocks for analog processing, such as two- and three-stage OpAmps, logarithmic photo detectors, or Gaussian computational circuits. It includes three capacitors, Cm1, Cm2 and Cc, of 100fF, 100fF and 5pF respectively. Details of the FPTA can be found in [15,19].

The evolutionary algorithm was implemented in a DSP that directly controlled the FPTA, together forming a board-level evolvable system with fast internal communication ensured by a 32-bit bus operating at 7.5MHz. Details of the EP were presented in [4]. Over four orders of magnitude speed-up of evolution was obtained on the FPTA chip compared to SPICE simulations on a Pentium processor (this performance figure was obtained for a circuit with approximately 100 transistors; the speed-up advantage increases with the size of the circuit). The evaluation time depends on the tests performed on the circuit. Many of the evaluation tests performed required less than two milliseconds per individual, which for example on a population of 100 individuals running for 200 generations required only 20 seconds. The bottleneck is now related to the complexity of the circuit and its intrinsic response time. SABLES fits in a box 8” x 8” x 3”.

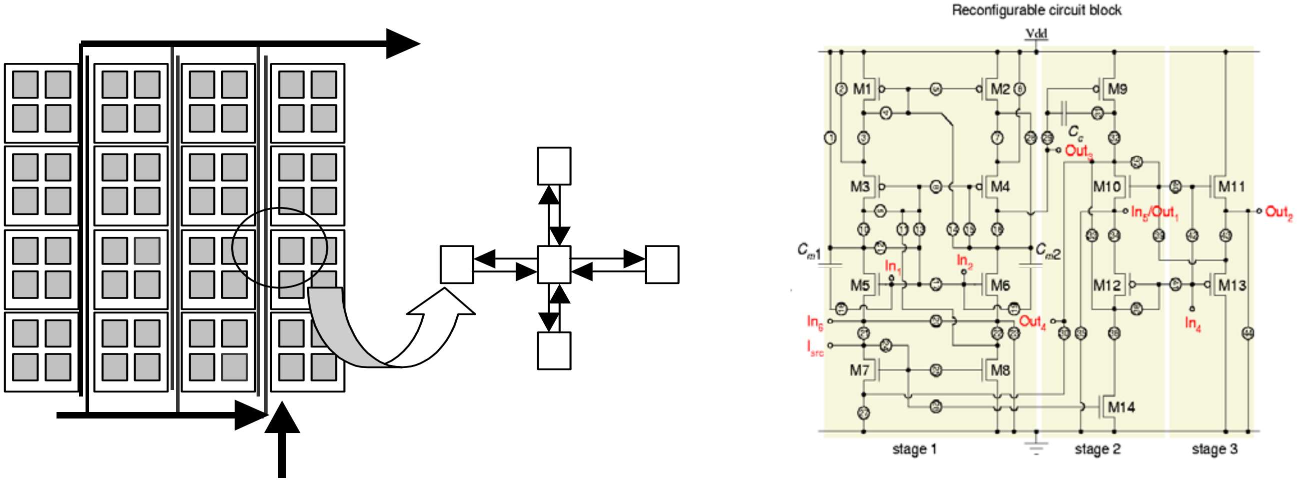

Figure 3. FPTA 2 architecture (left) and schematic of cell transistor array (right). The cell contains additional capacitors and programmable resistors (not shown).

2.2. An Evolution on SABLES

The following experiment illustrates an evolution on SABLES. The objective of this experiment is to synthesize a half-wave rectifier circuit. The testing of candidate circuits is made for an excitation input of 2kHz sine wave of amplitude 2V. A computed rectified waveform of this signal is considered as the target. The fitness function rewards those individuals exhibiting behavior closer to target (using a simple sum of differences between the response of a circuit and target) and penalizes those farther from it. After evaluation of 100 individuals, they are sorted according to fitness and a 9% portion (elite percentage) is set aside, the remaining individuals undergoing first crossover (70% rate), either among themselves or with an individual from elite, and then mutation (4% rate). The entire population is then reevaluated. In this experiment only two cells of the FPTA were allocated.

The left of Figure 4 depicts the waveforms for stimulus and response, the time allocated for stimulation and the time allocated for the GA in an evolutionary cycle. The right side is the detail, and illustrates the programming time of the new circuit and the stimulation with two periods of waveform, with two different looking responses for the two circuits being evaluated.

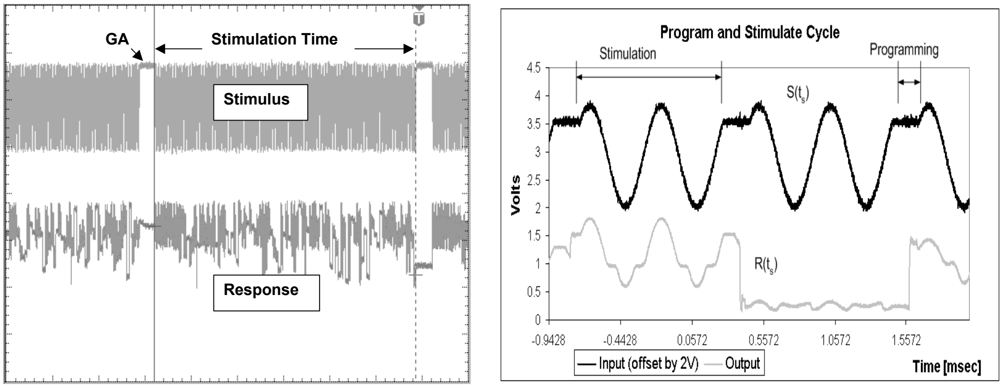

Figure 4. Stimulus-response waveforms during the evaluation of a population in one generation (left) and for 2 individuals in the population (right) A full GA cycle includes stimulus/response (113ms) and the generation of the next generation (6ms). The response was sampled at the maximum sampling rate of the on-board A/D (100kSamp/sec).

Figure 5 displays snapshots of evolution in progress, illustrating the response of the best individual in the population over a set of generations. The first caption shows the best individual of the initial population, while the subsequent ones show the best after 5, 50 and 82 generations. The solution, with a fitness below 4,500 is shown on the right. Figure 6 shows the convergence over a number of runs.

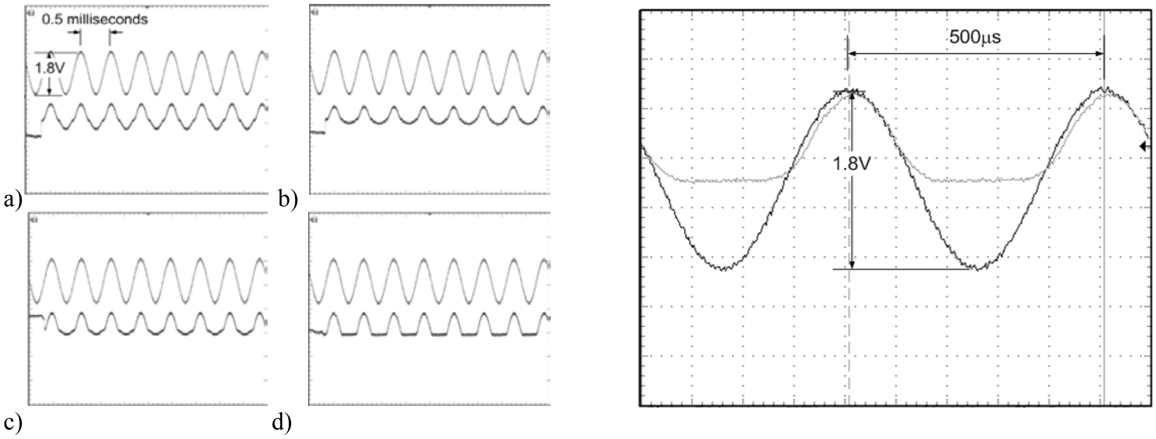

Figure 5. Evolution of a halfwave rectifier showing the response of the best individual of generation a) 1, b) 5, c) 50 and finally the solution at generation d) 82. The final solution, which had a fitness value less than 4500, is illustrated on the right.

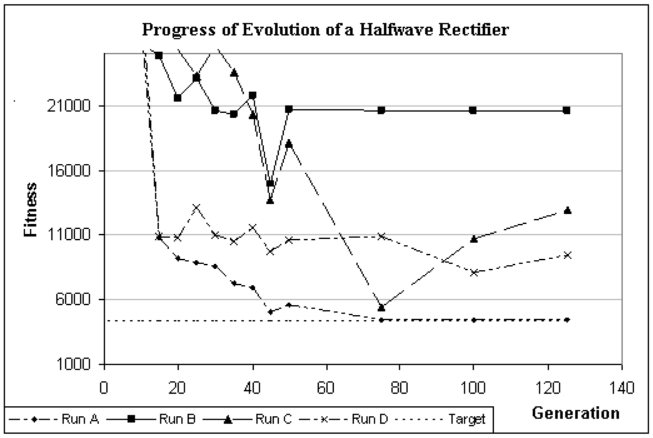

Figure 6. The fitness function as generations progress. The first few generations showed fitness values near 100,000 and are not shown on this scale.

3. On certain traps of evolutionary engineering

With SABLES enabling rapid evolvable hardware experiments, the focus has shifted from the hardware platform to algorithms. More specifically the focus became overcoming problems related to the formulation of requirements in a way that facilitate evolutions, and the translation of target specifications into the language of evolution, including representations, fitness function and parameters of the algorithm. One reason for the difficulty of evolving in autonomous systems is the necessity of providing complete up-front specifications, and one should emphasize that completeness may not often be obvious. In computer-assisted design the human can come back and provide extra information that may have been omitted in the beginning. That is not possible in an autonomous system, and evolution usually finds the easiest way to satisfy expressed requirements

-Transient solutions

The halfwave rectifier experiment provides examples of two situations in which evaluations of candidate solutions on the same hardware-in-the-loop may lead to highly-ranked individuals receiving high fitness function and yet when re-evaluated individually prove to have been only spurious solutions. Figure 7 illustrates both a transient behavior and a FPTA-state dependence. The transient behavior describes a configuration that is not stable as a function of time, whereas FPTA state dependence describes a configuration whose behavior depends on the previous configuration(s). Both of these behaviors are shown in Figure 7; most obviously, the function, which starts out looking similar to a halfwave rectifier ends up looking quite different. The transient behavior in this case occurred on a time-scale of about 1 second. Despite the transient behavior, the individual was selected as a solution because it was evaluated during a time-scale of about 2 milliseconds much shorter than the transient duration. In practice the transient behavior can be resolved by reevaluating the individuals for a longer time period.

The FPTA state dependence behavior occurred when the individual solutions programmed on the FPTA suffer somewhat from an apparent instability, which arises when the evaluation of a given individual depends on the previous state of the FPTA. The individual shown in Figure 7 was selected as a solution because, during the evaluation, its response must have matched quite well the expected function. However part a) of Figure 7 shows that the circuit does not behave sufficiently like the target rectifier, so the behavior exhibited in the evaluation must have been influenced by the previously downloaded configuration(s).

Parasitic as well as static capacitors in the chip explain this behavior. Capacitors can be charged during one configuration period and not completely discharged before the next configuration is tested, which leads to an undefined charge on capacitors and subsequently alters the behavior of the circuit. Nevertheless, for this particular experiment it has been observed that evolution weeds these individuals out and stable solutions are almost always found within the first 50 generations, or about 20 seconds

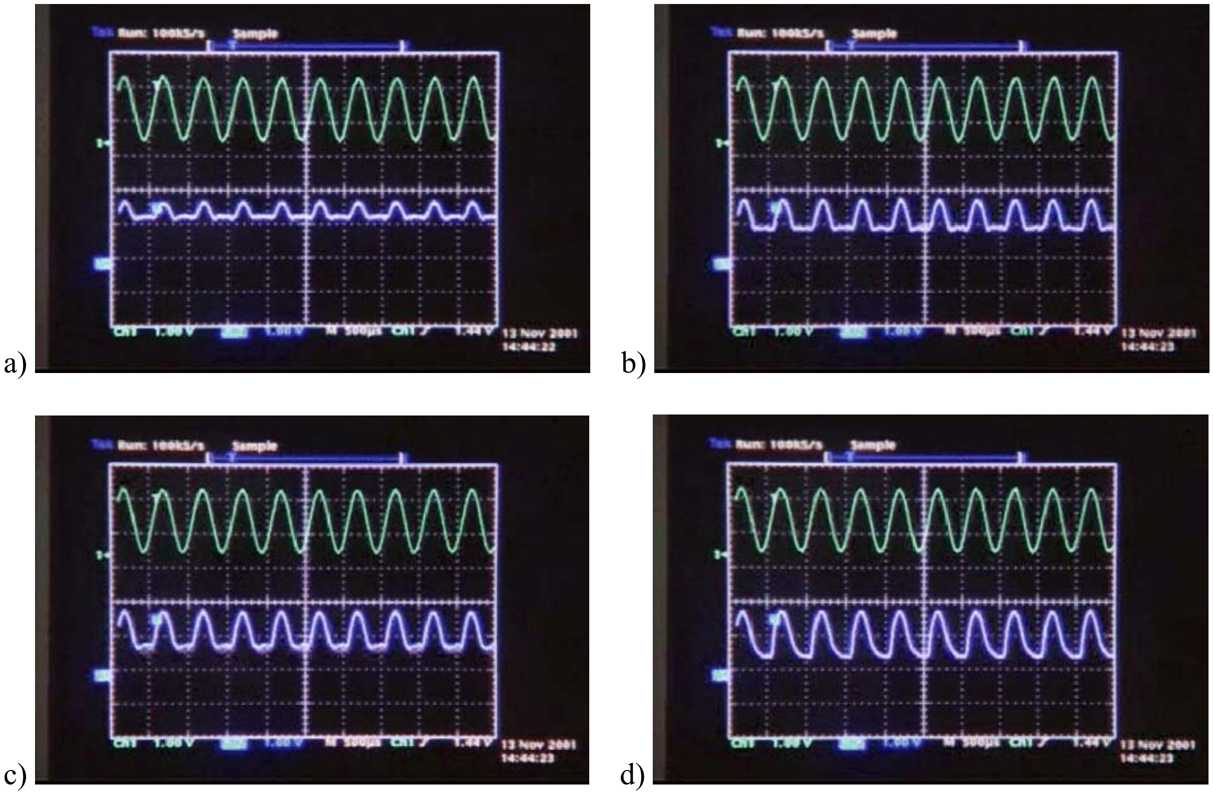

Figure 7. An example of transient behavior. The degradation shown from a) to d) occurred over the span of approximately 1 second.

.

The operational range of the evolved circuit in the frequency domain is another potential pitfall, since in principle the circuit behavior should be evaluated for the overall frequency domain in which it is expected to work. Figure 8 depicts half-wave rectifier response for different frequencies. From the graph at the left of the figure, it can be observed that the circuit works properly for the decade going from 500Hz to 5kHz, the frequency region in which the circuit was actually evolved. Nevertheless, the response deteriorates for higher frequencies, as illustrated by the graph at the right of Figure 8 for 50kHz.

These results are typical of a series of successful runs. Approximately 1 out of 10 runs ended with the algorithm getting stuck and not finding a solution at all for that small/fixed mutation rate.

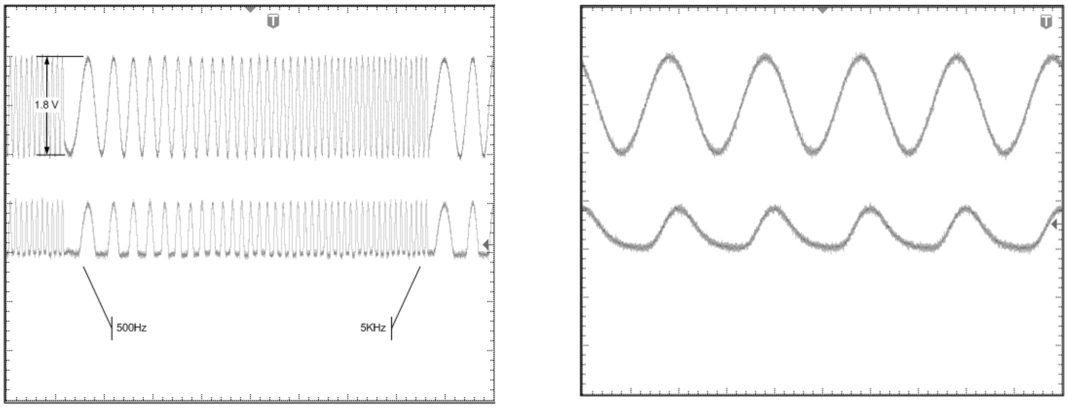

Figure 8. **Response of the half-wave rectifier for a frequency sweep from 500Hz to 5kHz(left). Deteriorated response at 50kHz.**

-Time constants

Some of the assumptions often implicit to human designers may be missing from the explicitly formulated requirements and thus from the fitness function of an evolutionary design. An example illustrated here is the implicit assumption that a logic gate should have the same behavior over a "frequency range" i.e. function with slow/DC signals as well as to faster input changing signals. The example illustrated here evaluated a circuit targeted as NAND gate providing input stimulus (using a SPICE transient analysis) with changes in the microsecond range. The correct behavior for this timescale was quickly achieved by evolution. However, an incorrect behavior was observed when the same circuit was simulated after evolution in the timescale of seconds. This is illustrated in Figure 9, where the first column shows the response of the deceptive circuit at the scale used in the fitness function (microseconds), and the second column the response of the same circuit when evaluated at a different timescale (seconds). Conversely, when the circuit is evaluated at a large timescale evolution often led to slow gates. The method applied to correct this situation was a derivative of the mixtrinsic evolution method introduced in [18]. In mixtrinsic evolution solutions for two types of models, which could be software or hardware, or as here models subject to different analysis can be obtained by either giving a combined fitness function for the two models or assigning the candidate solutions to one model or another during successive generations and thus letting evolution remove solutions that do not behave well on both models. In this particular case a two-transient analysis for each candidate circuit was performed during evolution, the first on a small timescale and the second on a larger timescale, solving the problem. For each circuit, the combined fitness measure was chosen to be the worse between the two evaluations, so that the genetic algorithm was driven to achieve a correct behavior at both timescales. The two right columns of Figure 9 show the response of a circuit evolved using this method with a correct response at both two timescales.

4. Summary and Conclusion

The paper presented a stand-alone board-level evolvable system (SABLES) and illustrated its performance with the evolution in seconds of a halfwave rectifier circuit. To date this is the fastest, most flexible and most compact stand-alone evolvable system for both analog and digital circuits. Intrinsic evolution using the same hardware-in-the-loop resources for consecutive evaluation of individuals may lead to transient solutions. In most cases these were eliminated simply by allowing extra time for evolution. Another possible trap may come from incomplete requirement specification such as those related to the timescale of operation of logic gates, in which case either slow or gates only operating on fast changing inputs may be obtained. These can be solved through mixtrinsic evolution, e.g .giving a combined fitness function to reflect desired behavior at two timescales.

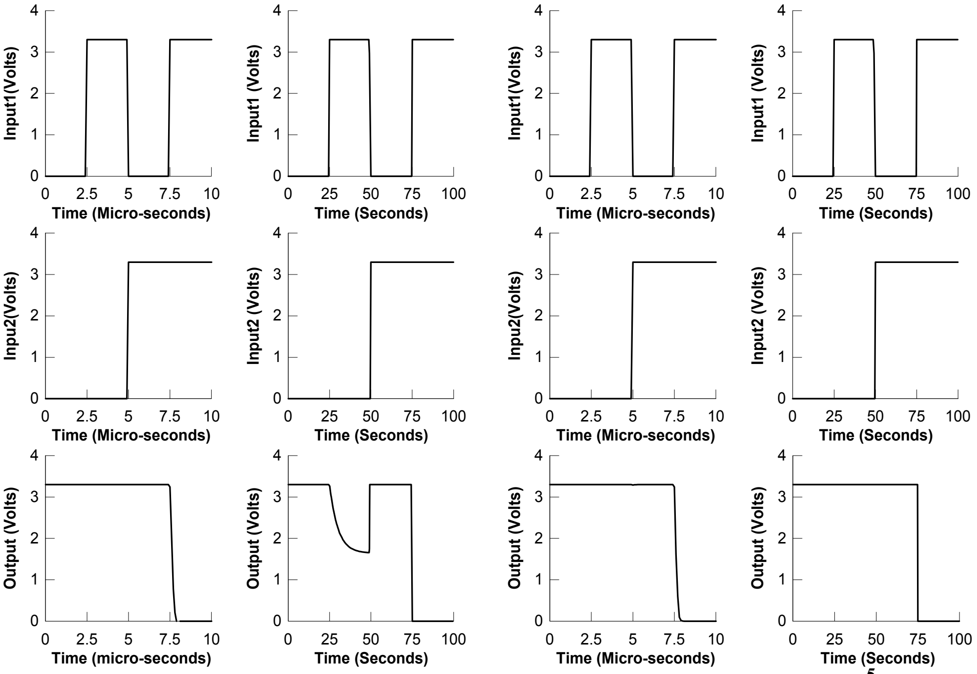

Figure 9. Evolved NAND gate evaluated in the timescale of microseconds (until 10^{-5} sec) shown in the first column. Incorrect behavior of the gate when it is simulated in the timescale of seconds (until 100 seconds) is shown in the second column. Evolved NAND gate using two different timescales (micro-seconds in the left and seconds in the right) show a correct behavior in columns three and four.

Acknowledgements

The research described in this paper was performed at the Center for Integrated Space Microsystems, Jet Propulsion Laboratory, California Institute of Technology and was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the National Aeronautics and Space Administration (NASA).

References

[1] J. R. Koza, F. H. Bennett III, D. Andre and M. A. Keane. Genetic Programming III – Darwinian Invention and Problem Solving, Morgan Kaufman, San Francisco, 1999..

[2] D. Keymeulen, G. Klimeck, R. Zebulum, A. Stoica, and C. Lazaro. "EHWPack: A Parallel Software/Hardware Environment for Evolvable Hardware". In Whitley Darrell (eds.), Proceedings of the Genetics and Evolutionary Computation Conference (GECCO-2000), July 8-12, 2000,pg.538-539. Las Vegas, Nevada USA. San Francisco, CA: Morgan Kaufmann.

[3] A. Thompson, "Silicon Evolution", Proceedings of Genetic Programming 1996 (GP96), J.R. Koza et. al. (eds), pp. 444-452, MIT Press, 1996.

[4] Ferguson, M.I., Stoica A., Zebulum R., Keymeulen D. and Duong, V. "An Evolvable Hardware Platform based on DSP and FPTA". To be published in Proceedings of the Genetic and Evolutionary Computation Conference, July 9-13, 2002, New York, New York.

[5] Shackleford B., Carter D., Snider G, Okushi E., Yasuda M., Koizumi H., Seo K., Iwamoto T., Yasuura H. "An FPGA-Based Genetic Algorithm Machine" FPGA 2000, Eighth ACM International Symposium on Field-Programmable Gate Arrays, Feb. 10-11, 2000.

[6] Higuchi T., Iwata M., Keymeulen D., Sakanashi H., Murakawa M., Kajitani I., Toda K., Salami M., Kajihara N., Otsu N. "Real-World Applications of Analog and Digital Evolvable Hardware". In IEEE Transactions on Evolutionary Computation. Vol. 3 Nbr. 3, September 1999, pp.220-235. Piscataway, NJ: IEEE Press.

[7] Koza, J., Yu, J., Keane, M. A.., and Mydlowec, W.,"Use of Conditional Developmental Operators and Free Variables in Automatically Synthesizing Generalized Circuits using Genetic Programming" Proceedings of the Second NASA / DoD Workshop on Evolvable Hardware, July 13 – 15 2000, Palo Alto, California. Pages 5 - 15.

[8] J. D. Lohn, S. P. Colombano, "Automated Analog Circuit Synthesis Using a Linear Representation", in Proc. of the Sec. Int. Conf. on Evolvable Systems:. M.Sipper, D.Mange and A. Pérez-Uribe (ed), vol. 1478, pp. 125-133, LNCS, Springer-Verlag, 1998.

[9] R.S. Zebulum, M. A. Pacheco, M. Vellasco, "Artificial Evolution of Active Filters: A Case Study", Proc. of the First NASA DoD Workshop on Evolvable Hardware, pp.66-75, IEEE Computer press., July, 1999.

[10] A. Thompson, P. Layzell, R. Zebulum, “Explorations in Design Space: Unconventional Electronics Design Through Artificial Evolution”, IEEE Trans. on Evolutionary computation, vol. 3, n. 3, pp. 167-196, 1999.

[11] G. Hollingworth, S. Smith, A. Tyrrel, “Safe Intrinsic Evolution of Virtex Devices”, In J Lohn et al.(eds.),Proceedings of NASA/DoD Workshop on Evolvable Hardware (EH2000), July 13-15,2000, (pp195-204). Palo Alto, CA, USA. IEEE Computer Society.

[12] A. Mesquita, F. Salazar, P.P. Canazio, “Chromosome Representation through Adjacency Matrix in Evolutionary Circuit Synthesis”, 2002 NASA/DoD Conference on Evolvable Hardware, Washington DC, USA, July , 2002 (to appear).

[13] Flockton and Sheehan, “Intrinsic Circuit Evolution Using Programmable Analog Arrays”, in Proc. of the Sec. International Conference on Evolvable Systems, M.Sipper, D.Mange and A. Pérez-Uribe (editors), vol. 1478, pp. 144-155, LNCS, Springer-Verlag, 1998.

[14] A. Stoica, “On Hardware Evolvability and Levels of granularity”, Proc. of the International Conference on Intelligent Systems and Semiotics 97: A Learning Perspective, NIST, Gaithersburg, MD, Sept. 22-25, 1997.

[15] A. Stoica, R. Zebulum, D. Keymeulen, R. Tawel, T. Daud, and A. Thakoor, Reconfigurable VLSI Architectures for Evolvable Hardware: from Experimental Field Programmable Transistor Arrays to Evolution-Oriented Chips. In IEEE Transactions on VLSI Systems, Special Issue on Reconfigurable and Adaptive VLSI Systems, vol. 9, No. 1,February 2001. (pp.227-232).

[16] A. Stoica, D. keymeulen, A. Thakoor, T. Daud, G. Klimech, Y. Jin, R. Tawel, V. Duong. Evolution of Analog Circuits on field Programmable Transistor Arrays. In J Lohn et al.(eds.),Proceedings of NASA/DoD Workshop on Evolvable Hardware (EH2000), July 13-15,2000, (pp.99-108). Palo Alto, CA, USA. IEEE Computer Society.

[17] Kajitani, I., Hoshino T., Kajihara N., Iwata M., Higuchi T. “An Evolvable Hardware Chip and its Application as a Multi-function Prosthetic hand controller”. In Proceedings of 16th National Conference on Artificial Intelligence (AAAI-99). American Association of Artificial Intelligence, 1999, pp. 182-187. Memlo Park, CA:IEEE Press.

[18] A. Stoica, R. Zebulum and D. Keymeulen. “Mixtrinsic Evolution”. In T. Fogarty, J. Miller, A. Thompson and P. Thompson, (eds.), In Proceedings of the Third International Conference on Evolvable Systems: From Biology to Hardware. (ICES2000). April 17-19, 2000, Edinburgh, UK. New York, USA, Springer Verlag. (pg.208-217).

[19] A. Stoica, R. Zebulum, and D. Keymeulen, “Progress and Challenges in Building Evolvable Devices”, Third NASA/DoD Workshop on Evolvable Hardware, Long Beach, July, 12-14, 2001, (pp.33-35), IEEE Computer Society.